Studi dan implementasi kriptografi pada keamananan sistem pengawasan

Catur Wirawan Wijiutomo– NIM : 13505020
Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15020@students.if.itb.ac.id

Abstrak
makalah ini membahas mengenai studi dan implementasi kriptografi untuk mengamankan sistem pengawasan. Sistem pengawasan yang dimaksud adalah sistem pengawasan untuk infrastruktur-infrastruktur TI seperti *router* dan *server*. Sistem ini memanfaatkan komunikasi jaringan untuk menghubungkan dengan yang diawasi dan umunya menggunakan arsitektur *client-server*. *Client* pada ssitem pegnawasan adalah suatu *node* pada jaringan seperti router, komputer atau perangkat yang diawasi oleh *server* yang memeriksa keadaan *client* secara periodik. pada sistem pengawasan ini terjadi pengiriman query ke server berupa keadaan node yang ingin dipantau. Sedangkan server adalah node yang melakukan pengecekan dan emngumpulakn hasil pengawasan ke dalam databse untuk kebutuhan analisis. Pengecekan yang paling sederhana adalah pengecekan keadaan node apakah sedang hidup atau mati. Dalam impelementasinya cukup banyak informasi penting yang dikirimkan ke server terkait versi sistem operasi, sisa ruang harddisk hingga informasi basis data.

Data pada sistem pengawasan biasanya tidak di enkripsi. Hal ini berawal pada awal implementasi sistem pengawasan yang hanya mengecek *node* yang berada pada jaringan yang terpisah dari jaringan publik. Namun pada perkembangannya *node-node* mulai tersebar dari jaringan yang tertup bahkan melewati dan berada pada jaringan publik. Padahal data-data ini berisi informasi yang cukup penting tidak boleh dibaca oleh pihak yang tidak bertanggung jawab. Karena data ini dapat menjadi modal dalam tahapan pengumpulan data untuk melakukan serangan jaringan.

Sistem monitoring saat ini yang umum diimplementasi memilki dua tipe yaitu berbasis SNMP, yang merupakan standar dalam sistem pengawasan dan berbasis agen, yang dibuat oleh masing-masing pembuat perangkat lunak. Pada makalah ini akan digunakan perangkat lunak *open-source* yaitu *nagios* sebagai studi sistem pegnawasan. Alasan pemilihan nagios yaitu source code yang tersedia, telah mendukung dua tipe sistem pengawasan yaitu SNMP dan agen, dan memberikan kebebasan kepada pengembang untuk membuat sendiri program pengecekan yang akan terhubung ke mdoul utama dari *nagios* tersebut.

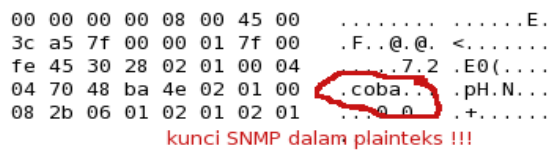
Kata kunci: SNMP, agen, pengawasan, *Nagios*, *UNIX*, *client-server*, kriptografi

1 Pendahuluan

pada saat ini jaringan komputer telah diterapkan dari skala usaha dari usaha kecil menengah hingga perusahaan besar. Dalam jaringan tersebut diltempatkan infrastruktur-infrastruktur penting yang memiliki fungsi signifikan dalam keberlangsungan usaha atau bisnis. Infrastrukutur tidak sepenuhnya robust dan dapat diandalkan. Dalam beberapa waktu dapat terjadi *downtime* yang berdampak pada berhentinya bisnsi yang menggunakan infrastrukut tersebut. Jika hal ini terjadi yang dibutuhkan adalah tindakan cepat dari administrator. Untuk menjamin berjalannya semua infrastruktur tersebut maka diimplementasikan sistem pengawasan untuk mempercepat diagnosis dan jika terjadi permasalahan dan tentu saja mempercepat aksi untuk menghindari kerugian yang lebih banyak .

Dalam implementasinya sistem pengawasan dibangun dengan arsitektur *client-server*. Pada umunya *client* mengirim data mengenai keadaan sistem ke server yang akan mengolah data tersebut sesuai permintaan dari server. Sesuai kebutuhan dari sistem pengawasan, data yang dipertukarkan bisa bermacam-macam antara lain *uptime*, kecepatan jaringan, sisa ruang harddisk, versi dari service hingga keadaan basis data. Data-data tersebut jika ditampilkan ke orang awam mungkin hanya sekedar data yang tidak berguna. Namun bagi peretas, data tersebut dapat digunakan untuk persiapan untuk melakukan sebuah serangan. Misalnya dengan menyadap data pengawasan yang tidak terenkripsi tersebut dengan menggunakan suatu program *sniffer* dapat diketahui versi dari suatu service, peretas selanjutnya akan mencari apakah terdapat lubang keamanan dari versi tersebut lalu melakukan eksploitasi terhadap lubang kemanaan yang ada pada versi tersebut.

Dari awal perkembangannya sistem pengawasan tidak memperhatikan faktor keamanan ini. Bahkan pada SNMP yang merupakan protokol standar untuk pengawasan bukanlah sebuah protokol yang didesain untuk keamanan. Hal ini ditambah dengan kenyataan data hasil pengawasan yang dialirkan melalui protokol SNMP merupakan plainteks yang dapat dilihat dengan mudah dengan suatu program *sniffer*. Seperti pada gambar kunci SNMP dapat mudah ditemukan dan berbasis plainteks.



Gambar 1: kunci yang tidak terenkripsi pada SNMP

Perkembangan terbaru dari SNMP telah dilengkapi keamanan namun versi terbaru ini belum secara penuh dideploy bahkan untuk mesin-mesin lama hal ini cukup sulit untuk diimplementasikan.

Keadaan ini tentu saja cukup memprihatinkan dengan perekembangan kakas hack yang lebih mudah digunakan bahkan user yang tidak berpengalaman akan mampu melakukan sniffing. sehingga dalam makalah ini dikembangkan sebuah teknik untuk mengimplemtasikan pengamanan data pengawasan dengan menggunakan kriptografi.

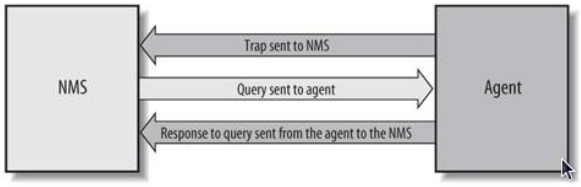
Tujuan utama dari implementasi ini adalah menemukan suatu solusi dengan kriptografi yang tepat untuk aliran data pengawasan dalam jaringan. Tentu saja selain keamanan yang terjamin solusi tersebut harus memilki kecepatan dan kemudahan implementasi. Hal ini dilihat dari kenyataan bahwa dalam beberapa kasus digunakan periode yang cepat untuk pengawasan yaitu permenit, proses kriptografi juga tidak boleh memberatkan perangkat sehingga menganggu fungsi dari perangkat tersebut dan dalam implementasinya data yang terkriptografi tidak boleh terlampau besar karena akan memberatkan jaringan.

2 Tipe dan model komunikasi sistem pengawasan

Sistem pengawasan diimplementasikan dalam model klien dan server. Pada kasus jaringan umum server pengawas berada pada jaringan yang terhubung dengan node-node yang diawasi.

Dalam sistem pengawasan terdapat dua buah entitas yang membentuk sistem yaitu manager dan agen. Manager adalah server yang menjalankan suatu sistem perangakt lunak yang dapat menangani tugas manajemen pengawasn untuk jaringan. Manager ini juga disebut sebagai NMS (Network Management stations). sebuah NMS bertanggung jawab untuk mengumpulkan dan menerima query dari agent-agent yang terdapat dalam jaringan.

Entitas selanjutnya adalah agen, merupakan perangakt lunka yang terpasang di sisi client yang diawasi. Secara sederhana agen merupakan program socket client sederhana yang menghubungi program socket server pada NMS.



Gambar 2: komunikasi NMS dan agen

Dalam mengimplemntasikan agen yang berbasi socket digunakan protokol TCP atau UDP. Keduanya memiliki kelemahan dan kelebihan masing-masing. Pemilihan ini penting karena dapat mempengaruhi algoritma kriptografi yang akan diimplemtnasikan. Untuk lebih jelas tmengenai hubungan NMS dan agernt dapat digambarkan:

UDP adalah protokol yang sebeanranya unreliable. Namun dalam beberapa kasus jaringan merupakan protokol yang lebih baik. Hal ini disebabkan sifatnya yang cepat. Kekurangan utama yang menyebabkan protokol ini tidak cocok pada implementasi keamanan dengan kriptografi adalah tidak terjaminnya urutan paket yang akan diterima di sisi penerima. Hal ini tentu saja menyebabkan salah satau jenis kriptografi seperti block cipher yang menggunakn CFC tidak dapat diimpelmtasikan. Dari infromasi ini kita menyadari mengapa versi awal dari SNMP tidak menggunakan kriptografi karean protokol ini menggunakan UDP untuk pengiriman datanya.

TCP secara umum memilki sifat yang bertolak belakang dengan UDP. Namun sambungan TCP sekali-kali terjadi loss. Namun dari dua pilihan protokol transport TCP memberikan layanan yang lebih baik untuk data yang terkriptografi. Sisi buruk yang dapat kita perkirakan adlah pembukaan port. Hal ini sebenarnya membuka sebuah celah sehingga dalam impelentasinya selain harus mengenkripsi pesan juga harus melindungi port yang terbuka untuk pertukaran data pengawasan.

Permasalahan port dapat ditangani dengan menggunakan firewall baik pada sisi klien dan server. Untuk lebih mengingkatkan fleksibilitas dan mempermudah konfgiurai dapat digunakan TCP wrapper.

3 kriteria kriptografi untuk NMS

Pada model sistem pengawasan terdapat kriteria tertentu yang harus dipenuhi antara lain:

1. kecepatan proses enkripsi dan dekripsi yang cepat

   kecepatan proses enkripsi dan dekripsi diperlukan karena pada beberapa kasus pengawasan digunaakn tenggat waktu yang cukup singkat dari satu pengecekan ke pengecekan berikutnya. Normalnya digunakan interval waktu 5 menit namun dalam kasus tertentu ksiarannya antara 30 detik – 60 detik.

2. resource komputasi yang dipakai dalam proses enkripsi dan dekripsi relatif kecil

   dalam beberapa kasus suaut server melakukan pengecekan pada node yang jumlah cukup banyak. Dqalam kasus tanatp enkripsi hal tersebut sudah memberikan beban CPU yang tinggi kepada server pemeriksa atau klien sehingga menganggu fungsi aslinya.

   Terlebih enkripsi akan diimplentasikan dalam setiap pesan sehingga jika menggunakn enkripsi yang menggunakan tingkat komputasi yang tinggi tetnu saja akan memakan banyak resource dan mengurangi kinerja infrastruktur. Penambahan kecepatan prosessor atau memroty tentu bukan menjadi opsi proses kriptogrfai harus menggunakan resource yang kecil.

3. Hasil enkripsi yang relatif kecil

   dalam beberap kasus enkripsi akan

menggembungkan ukuran dari file asli. Dalam implementasi protokol NMS hal ini sebisa mungkin dihindari dan diminimalkan karena NMS mengirimkan data ke dalam jaringan dalam interval waktu yang cukup singkat. Sehingga jika paket yang dikirimkan terlampau besar jsutru akan membebani jaringan.

4. mudah untuk diimplemtasi

   kriteria ini mentikaberatkan pada mekanisme aplikasi kriptografi yang digunakan pada program. Jika terlalu rumit justru akan menyulitkan proses adminitrasi.

sehingga jenis dan algoritma kriptograf yang dipilih untuk diimplementasikan harus memberikan sisi keamanan yang tidak menghilangakn kriteria dari fungsi pengawasan itu tersebut.

Dengan dipilihnya TCP akan memberikan kita fleksibilitas untuk algoritma kriptiografi karena TCP menajmin data yang diterima memilki urutan yang sesaui sehingga kompleksitas dari kriptografi yang dapat diterapkan emnajdi tidak terbatas sehingga tipe dan mode kriptografi tidak menjadi kriteria dalam enkripsi protokol NMS.

3.1.1 Tpe dan mode kriptografi untuk NMS

Tipe kriptografi yang menjadi calon dalam enkripsi pesan sistem pengawaasan adalah kriptografi kunci simetri dan kriptografi kunci publik. Keduanya memiliki kekurangan dan kelebihan masing-masing.

kelebihan kriptogafi kunci simetri adalah aplikasi kunci simetri dapat di desain untuk aplikasi yang membutuhkan data throuput yang cepat hal ini. Dibandingkan dengan algoritma kunci publik yang lambat dalam proses enkripsi dan dekripisi Kelemahannya adalah manajemen kunci jika dibandingkan dengan kunci publik yang dapat kita atur kunci publik dan kunci private-nya.

Sehingga dari kriteria protokol NMS dan melihat ciri dari masing-masing teknik kriptografi.teknik yang cocok digunakan adalah kriptografi kunci simetri.

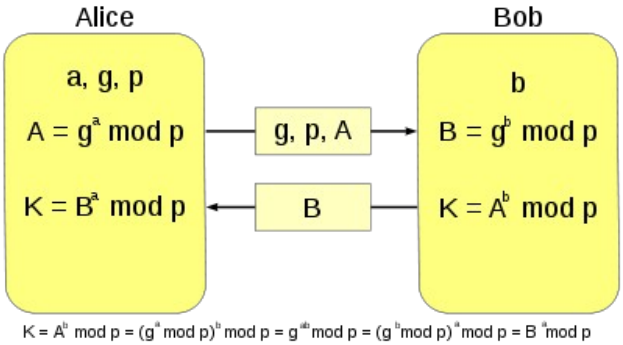
4 Perangkat Lunak kriptografi

Pada nagios terdapat sebuah agen yang dikembangka oleh komunitas nagios yang disebut NRPE. Pada NRPE ini sudah menggunakan enkripsi yang menggunakan library SSL. Pastikan pada saat mengkompilasi NRPE menggunakan flags –enable-ssl. Sehingga komunikasi NRPE akan mengggunakan enkripsi. Contohnya pada terminal saat instalasi dengan perintah
#./configure –enable-ssl
#make
#make install
library SSL/TLS ini menyediakan beberapa fungsi enkripsi. NRPE menggunakan fungsi Anonymous Diffie Helman(Anon-DH) dengan SHA1 dan enkripsi AES-256 bit.

Anon-DH merupakan penerapan enkripsi kunci publik namun tidak memerlukan pembuatan kunci yang digenerate sebelumnya untuk melakukan komunikasi. Setiap agen dan NMS mempertukarkan parameter DH. Yaitu bilangan prima Y dan basis P. yang membedakan dengan DH yang biasa adalah pada Anon-Dh pertukaran ini tidak digunakan authentifikasi ataupun sertifikasi pesan.

Pertukaran kunci Diffie-Helman mengizinkan 2 pihak yang tidak bertemu untuk mempertukarkan pesan secara aman dalam jalur komunikasi yang tidak aman. Secara umum D-H digunakan untuk mempertukarkan kunci yang digenerate secara acak.

skema dari D-H bergantung pada fungsi satu arah yaitu Y^x mod P, nilai Y dan P disetujui dai awal oleh kedua belah pihak. Dalam NRPE nilai ini otomatis ditentukan. Contohnya dengan menentukan Y=7, yang merupakan bilangan sebagai generator publik dan P=11. Sebagai bilangan generator prima. Kedua belah pihak dalam contoh disebut alice dan bob



Gambar 3: protokol D-H

langkah 1:
alice: alice memilih nomor secara acak misalnya 3 dan diberi label a
bob: bob memilih nomor secara acak misalnya 6 dan diberi label b
langkah 2:
alice: alice menaruh 3 ke dalam fungsi satu arah Y^a mod p = 7^3 mod 11 = 343 mod 11 = 2
Bob: bob meproses 6 dalam fungsi satu arah Y^b mod p = 7^6 mod 11 = 117.649 mod 11 =4
langkah 3
alice : alice mengirimkan hasil nilai 2 ke bob
bob: bob mengirimkan hasil nilai 4 ke alice
langkah 4:
alice : alice menerima hasil dari bob dan melakukan fungsi satu arah lagi, $(y^b \text{ mod } p)^a$ mod p = 4^3 mod 11 = 64 mod 11 = 9
bob : bob menerima hasil dari alice dan melakukan fungsi satu arah lagi $(y^a \text{ mod } p)^b$ mod p = 2^6 mod 11 = 64 mod 11 =9
hasil dari langkah 4 merupakan kunci dari cipherteks

penggunan anonymous-DH sebenarnya telah memberikan keamanan untuk keadaan minimal dua pihak berkomunikasi. namun tidak adanya authentifikasi untuk pertukaran parameter DH dan bilangan a dan b. membuat DH lemah terhadap serangan penyadapan.

Skenario penyadpaan akan dijelaskan dengan keadaan

sebagai berikut. Misalkan eve adalah penyadap. Akan tetapi eve tidak mengubah isi dari komunikasi mereka. Misal:
s = kunci rahasia yang dibagikan. s =2
a = kunci private alice, a =2
b = kunci provate bob, b = 15
g = publik base g =5
p = publik prima p =23
maka jalan dari skenario penyadapan dapat dijelaskan dalam tabel-tabel sebagai berikut:
langkah alice:

Alice	
tahu	Tidak tahu
P = 23	b =15
Basis g = 5	
A = 6	
5^6 mod 23 =8 (dikirm ke bob)	
5^b mod 23 =19 (diterima dari bob)	
19^6 mod 23 =2	
8^b mod 23 =2	
19^6 mod 23 = 8^b mod 23	
S = 2	

Langkah bob :

Bob	
tahu	Tidak tahu
P = 23	a =6
Basis g = 5	
B =15	
5^{15} mod 23 =19 (dikirm ke alice)	
5^a mod 23 =8 (terima dari Alice)	
8^{15} mod 23 =2	
19^a mod 23 =2	
8^{15} mod 23 = 19a mod 23	
S = 2	

Dalam menyadap eve akan mendapatkan g, p, a dan b dari jalur komunikasi dan mencoba nilai a dan b yang mungkin sehingga didapat angka yang sama untuk kunci a dan b.
langajh eve:

eve	
tahu	Tidak tahu
P = 23	a =6
Basis g = 5	b =15
5^a mod 23 =8	
5^b mod 23 = 19	
19^a mod 23 = s	
8^b mod 23 = s	
19^a mod 23 = 8^b mod 23 (dilakukan brute force)	

Dari mekanisme DH tersebut sebenarnya kriteria untuk kemudahan implementasi telah dipenuhi. Antara dua node yang berkomunikasi tidak harus melakukan distribusi key.

akan tetapi kelemahan terhadap penyadap ini akan dicoba untuk diperbaiki dengan mengenkripsi parameter DH dengan kriptografi lain yang lebih aman. Implementasi yang dilakukan dengan meningkatkan protokol DH dengan menambahkan kunci sesi yang terenkripsi dengan RC4

RC4 yang digunakan adalah stream cipher kunci simetris yang cukup cepat. Kriptografi ini mengizinkan hingga panjang kunci 2048 bit. Dan menggunakan tabel internal yang dibentuk dari kunci sehingga dapat digunakan ukuran kunci yang kecil. RC4 ini mulanya didesain oleh RSADSI(*RSA Data Security inc)* dan cipher ini merupakan rahasia dagang dari RSADSI yang tidak dapat digunakan secara bebas. Tetapi orang tidak dikenal mendistribusikan kode ke internet pada tahun 1994. sehingga saat ini RC4 sudah bukan rahasia dagang lagi sehingga cipher ini dapat digunakan dengan bebas tanpa harus membayar biaya sepeser pun.

implementasi dari perangkat lunak dapat dengan menggunakan bahasa perl, kita namakan program dengan dh.pl, berikur source codenya:

```
#!/usr/local/bin/perl
($g,$e,$m)=@ARGV,$m||die"$0 gen exp mod\n";
print`echo "16dio1[d2%Sa2/d0<X+d
*La1=z\U$m%0]SX$e"[$g*]\EszlXx+p|dc`
```

Selain itu ditulis juga program cipher RC4 yang dinamakan rc4.pl yang akan digunakan untuk mengenkripsi salah satu parameter DH.

```
#!/usr/bin/perl -p
INIT{sub Q{$s[($_[0]+=$_[1])%=256]}sub
S{@s[$y,$x++]=@s[$x,$y]}@k=pop=~/../g;
S$y=map{SQ$y,$_+hex$k[$x%@k]}@s=0..255}s/\C/
$&^chr Q S Q$y,Q$x/eg
```

Untuk menggunakan program rc4.pl dengan memberikan kunci dalam bentuk hex dan program akan mengenkripsi standar input ke standar output. Pada contoh ktia mengenkripsi pesan berisi “test message” dengan 32 bit hex key “12abcdef” pada *command line*.
$ echo test message | rc4 12abcdef > test.rc4
untuk mendekripsi pesan dapat dilakukan dengan cara:
$ rc4 12abcdef < test.rc4

pertama sekali kita pilih bilangan bilangan untuk public generator g =3 , publik prima P=10001 dan x= 9a2e(hex)
>dh $g $P $a
akan menghasilkan bilangan hexa c366
angka g, P dan c366 ini dikirimkan ke agen. Dan digunakan untuk dekripsi.

Selanjutnya dibuatlah kunci sesi yang dienkripsi RC4 menggunakan pertukaran kunci D-H. Session key dibangkitkan dari pemilihan bilangan secara acak misalnya 4c20.
> dh c366 4c20 $P

maka hasilkunci sesi nya adalah bilangan hex ded4, selnajutnya sebut sebagai s.

selanutnya untuk mengirmkan session s key dengan melakukan perhitungan ini di agen.
> dh $g 4c20 10001

maka b = 6246, nilai ini juga dikirimkan ke server. Sehingga server dapat mendapatkan nilai kunci sesi dengan menggunakan program:
> dh 6246 9a23e $m
program akan menghasilkan session key yang sama sperti pada agen. Sehingga server dan agen memiliki session key yang sama. Dengan session key inilah data dienkripsi. Hal ini dapt duji dengan perintah di command line sebagai berikut:
> cat msg | rc4.pl ded4 | uuencode r r | mail hostname

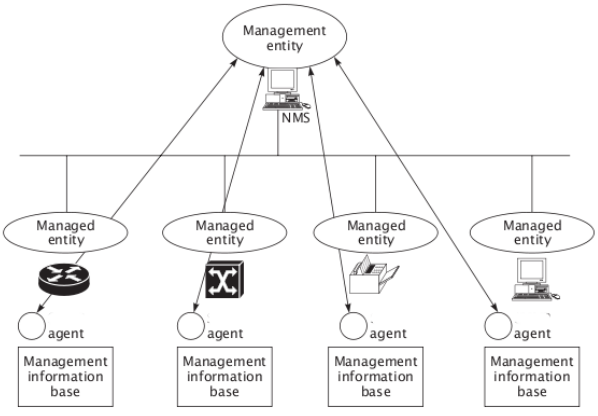
hal ini bekerja karena :
$(g^a)^b \bmod P = (g^b)^a \bmod P$
dan

$A = g^a \bmod P$
$B = g^b \bmod P$

sehingga:
$s = A^a \bmod P = Bb \bmod P$

langkah selanjutnya adalah dibuat program socket sederhana untuk server dan client. Program soket server diletkkan di server NMS dan program klien diletakkan di node-node yang diawasi. Program server akan memprose query dari client yang isinya berupa data hasil pengawasn.

Pada program tersebtut pada salah sstu prosedurnya digunakan program rc4.pl dan dh.pl untuk mengenkripsi pesan. Pastikan keluaran dari program client dan server tersebut sesuai dengan output yang didukung oleh modul utama dari nagios sehingga dapat dipeoses ke database.



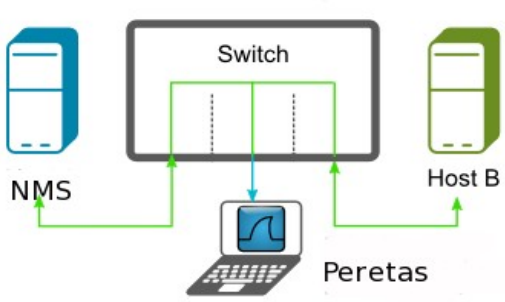
Gambar 4: contoh sistem NMS

5 Pengujian

5.1.1 Rancangan tes pengujian

Pengujian dilakukan dengan menggunakan server yang terpasang software nagios yang akan menerima masukan dari agen yang menggunakan kriptografi. Server akan melakukan pengawasan terhadap beberapa klien.

Untuk pengujian Topologi jaringan ethernet IEEE 802.3 yang digunakan cukup sederhana. Dengan menggunakn topologi star dengan infrastruktur *switch*. *Switch* menghubungkan server NMS, host dan peretas berada dalam satu jaringan. Dengan berada pada astu jaringan memungkin peretas untuk melakukan serangan sniifer dengan mode *prosmicious* atau melakukan *ARP poisoning* yang akan menyebabkan semua paket tertangkap.



Gambar 5: Jaringan Pengujian

Pada jaringan tersebut akan dipasang sebuah snifffer yang akan mendeteksi paket dari server ke klien dari sini akan dilihat apakah sniffer dapat membaca paket yang terkirim sudah tersamarkan oleh kriptografi atau belum.

5.1.2 Hasil pengujian

Dari hasil pengujian dengan menggunakn program sniffer wireshark informasi yang dikirimkan antara NMS dan agen sudah tidak berupa plainteks. Dan informasi kunci sesi dan parameter DH dikirimkan dengan skenario menggunakan jalur komunikasi yang aman.

Data yang dikirim berupa hasil pengecekan load average:
OK - load average: 0.20, 0.15, 0.15| load1=0.200;15.000;30.000;0;
load5=0.150;10.000;25.000;0;
load15=0.150;5.000;20.000;0;

dari hasil sadap terjadi pertukran data sebanyak 16 kali. Beberapa adalah aksi handshake dari protokol TCP yang digunakan.

Hasil sniffer:
Data:1503010020F0FF528E16B2ED789D19391467F735B61972E0...
Data:160301004601000042030149D4A3D1C5B70491F185957801...
Data:160301004610000042004066 73B91B4D0CAD8387343702AF...
Data:170301002085C0C9C44DA60598521F0F27D6977CB3CE46EF...
data sudah tidak berupa plainteks. Kriteria keamanan telah dipenuhi pada pengujian ini.

performa kecepatan pengiriman data setara dengan kecepatan RTT(*Round Trip Time*) yang diperoleh dari aksi ping. Hal ini juga dipengaruhi oleh *congestion* yang terjadi pada jaringan dan besara data yang dikirimkan.
Hasil sniffer kecepatan data dengan melihat kecepatan

paket ack:
The RTT to ACK the segment was: 0.000035000 seconds
The RTT to ACK the segment was: 0.000373000 seconds
The RTT to ACK the segment was: 0.000034000 seconds
The RTT to ACK the segment was: 0.000219000 seconds
The RTT to ACK the segment was: 0.001947000 seconds
The RTT to ACK the segment was: 0.000257000 seconds
sehingga waktu total pengiriman data adalah 0.0002865000 detik. Kriteria paket yang kecil dapat kita lihat dari kecepatan pengiriman data tersebut.

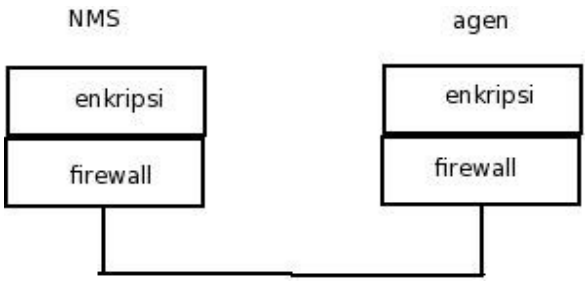
Dari resource komputasi, baik NMS atau host B tidak memperoleh load 1 menit yang melebih angka 0.1 setiap kali proses pengecekan dilakukan.

6 Kesimpulan

Aliran data yang aman antara server NMS dan node-node merupakan tujuan yang ingin dicapai dalam makalah ini. Enkripsi merupakan salah satu cara dalam mengamankannya. Tipe keriptografi yang dijelasakn dan diimplemtasi dalam makalah ini merupakan cara pengamanan aliran data dengan tetap memperhatikan kriteria untuk aliran data pada NMS.

Kriptografi merupakan proses yang membutuhkan reosurce komputasi yang cukup besar. Untuk mengamankan pesan dengan lebih baik tentu saja akan membutuhkan komputasi yang besar. Untuk beberapa kasus khusus hal ini adalah harga yang sesuai untuk dibayar. Namun untuk sistem pengawasan hal ini sebaiknya dihindari. alasan utama adalah dilibatkannya banyak node dalam jaringan dan dilakukan pengecekan berulang-ulang dengan interval waktu yang relatif singkat. Sehingga implemtasi dengan kriptografi teraman saat ini yang menggunakan komputasi besar tentu saja dapat menghabiskan resource dan menganggu fungsi dari node-node tersebut.

Teknik Kriptografi D-H telah memenuhi kriteria dalam pengamanan aliran data pada sistem pengawasan hal ini juga telah ditingkatkan dengan menggunakn kunci sesi yang dienkripsi dengan teknik RC4.



Gambar 6: keamanan dengan firewall dan enkripsi

Dalam dunia nyata menggunakan enkripsi untuk sistem pengawasan tidak menjadi solusi yang menyelesaikan maslaah keamaan data pengawasan. Penyadapan dapat terjadi di manapun. Kakas yang digunakn untuk serangan juga semakin mudah digunakan. Sehingga perlu digunakan keamanan berlapis yang melibatkan *layer-layer* dalam konsep *7 OSI-Layer.* Untuk lebih meningkatkan keamanan, teknik enkripsi ini sebaiknya dipadukan dengan *firewall* yang akan menyaring aliran data sehingga data hanya dialirkan ke node-node yang berhak.

7 Daftar Pustaka

Josephsen,David, *Building Monitoring Infrasctructure with nagios. Prentice hall*. 2007

TurnBull, james. *Pro Nagios 2.0.* Apress. 2006

Mauro, Douglas. *Essential SNMP, 2nd Edition.* O'reilly. 2005.

Schneier, Bruce. *Applied Cryptography 2nd Edition protocols, algorithm and source code in C.* John Willey & sons. 1996

Menezes, A, P. Van Ooschot, dan S. Vanstase. *handbook of Applied Cryptography.* 1996

Adam back. Diffie halman in 2 line of perl. http://www.cypherspace.org/adam/rsa/rc4.html

Adam back. Perl version of RC4. http://www.cypherspace.org/adam/rsa/perl-dh.html